**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В. И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «WEB-технологии»**

Тема: **ТЕТРИС НА JAVASCRIPT**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1381 |  | Возмитель В. Е. |
| Преподаватель |  | Беляев С. А. |

Санкт-Петербург

2023

## Цель работы.

Целью работы является изучение работы *web*-сервера *nginx* со статическими файлами и создание клиентских *JavaScript* *web*-приложений.

**Основные теоритические сведения.**

Асимметричные ключи используются в асимметричных алгоритмах

шифрования и являются ключевой парой. Закрытый ключ известен только

владельцу. Открытый ключ может быть опубликован и используется для

проверки подлинности подписанного документа (сообщения). Открытый

ключ вычисляется, как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.

По секретному ключу можно вычислить открытый ключ, но по открытому ключу практически невозможно вычислить закрытый ключ.

nginx (https://nginx.ru/ru/) – веб-сервер, работающий на *Unix*-подобных

операционных системах и в операционной системе *Windows*.

JavaScript (https://developer.mozilla.org/ru/docs/Web/JavaScript) – язык

программирования, он поддерживает объектно-ориентированный и функциональный стили программирования. Является реализацией стандарта *ECMAScript*.

## Задание.

Необходимо создать web-приложение – игру в тетрис. Основные требования:

– сервер – *nginx*, протокол взаимодействия – *HTTPS* версии не ниже 2.0;

– отображается страница для ввода имени пользователя с использованием *HTML*-элементов <*input*>;

– статическая страница отображает «стакан» для тетриса с использованием *HTML*-элемента <*canvas*>, элемент <*div*> используется для отображения следующей фигуры, отображается имя пользователя;

– фигуры в игре – классические фигуры тетриса (7 шт. тетрамино);

– случайным образом генерируется фигура и начинает падать в «стакан».

– пользователь имеет возможность двигать фигуру влево и вправо, повернуть на 90 градусов и «уронить»;

– если собралась целая «строка», она должна исчезнуть;

– при наборе некоторого заданного числа очков увеличивается уровень, что заключается в увеличении скорости игры;

– пользователь проигрывает, когда стакан «заполняется», после чего ему отображается локальная таблица рекордов;

– вся логика приложения написана на *JavaScript*.

Постарайтесь сделать такую игру, в которую вам будет приятно играть. Помните, когда-то эта игра была хитом! Преимуществом будет использование звукового сопровождения событий: падение фигуры, исчезновение «строки».

**Выполнение работы.**

1. Были сгенерированы открытый и закрытый ключи для использования шифрования.
2. Настроен сервер *nginx* для работы по протоколу *HTTPS*.
3. Разработан интерфейс пользователя.
4. Написана логика программа

4.1 Обеспечен ввод имени пользователя

4.2 Обеспечено создание всех фигур, их движение по таймеру

4.3 Настроено управление пользователем фигурой

4.4 Подключена таблица рекордов

Описание созданных сущностей:

1. Класс *Game* – предназначен для хранения всей логики игры: игровое поле, все фигуры, их поворот, движение, столкновения и тд.

Реализуются следующие методы класса:

* *moveLeft()* – движение в левую сторону
* *moveRight()* – движение в правую сторону
* *moveDown()* – движение вниз
* *rotateBlocks()* – поворот блоков фигуры
* *IsBarrier()* – метод, проверяющий столкновение фигуры с чем-либо
* *lockPiece()* – метод фиксирует фигуру на игровом поле
* *createPlayField()* – метод, создающий игровое поле
* *createPiece()* – метод для создания фигуры
* *getState()* – метод, созданный для получения всей необходимой информации о классе *Game*
* *clearlines()* – метод для удаления полностью заполненной строчки
* *updateScore()* – метод, обновляющий статистику игры (очки, уровень)

1. Класс View – предназначен для отображения логики на экране.

Реализуются следующие методы класса:

* *viewRender()* – общий метод, связывающий следующие методы
* *viewPLayfiled()* – метод, предназначенный для отрисовки игрового поля
* *viewBlock()* – метод, отображающий прямоугольник на экране ( часть фигурки)
* *viewNextBlock ()* – метод, отображающий следующую фигуру

1. Класс *Recorder* – класс, описывающий таблицу рекордов, хранит информацию об игре: имя пользователя, рекорд, место в таблице.

Реализован следующий метод:

* *writeResults()* – метод, добавляющий в общую локальную таблицу рекордов текущий результат.

1. Класс *Controller* – предназначен для управления игрой.

Реализуются следующие методы класса:

* *handleKeyDown()* – реализация switch технологии для отслеживания сигналов с клавиатуры
* *boostSpeed()* – метод, увеличивающий скорость при увеличении уровня игры.
* *checkLvl()* – вспомогательный метод, проверяющий изменение уровня игры.

**Интерфейс.**

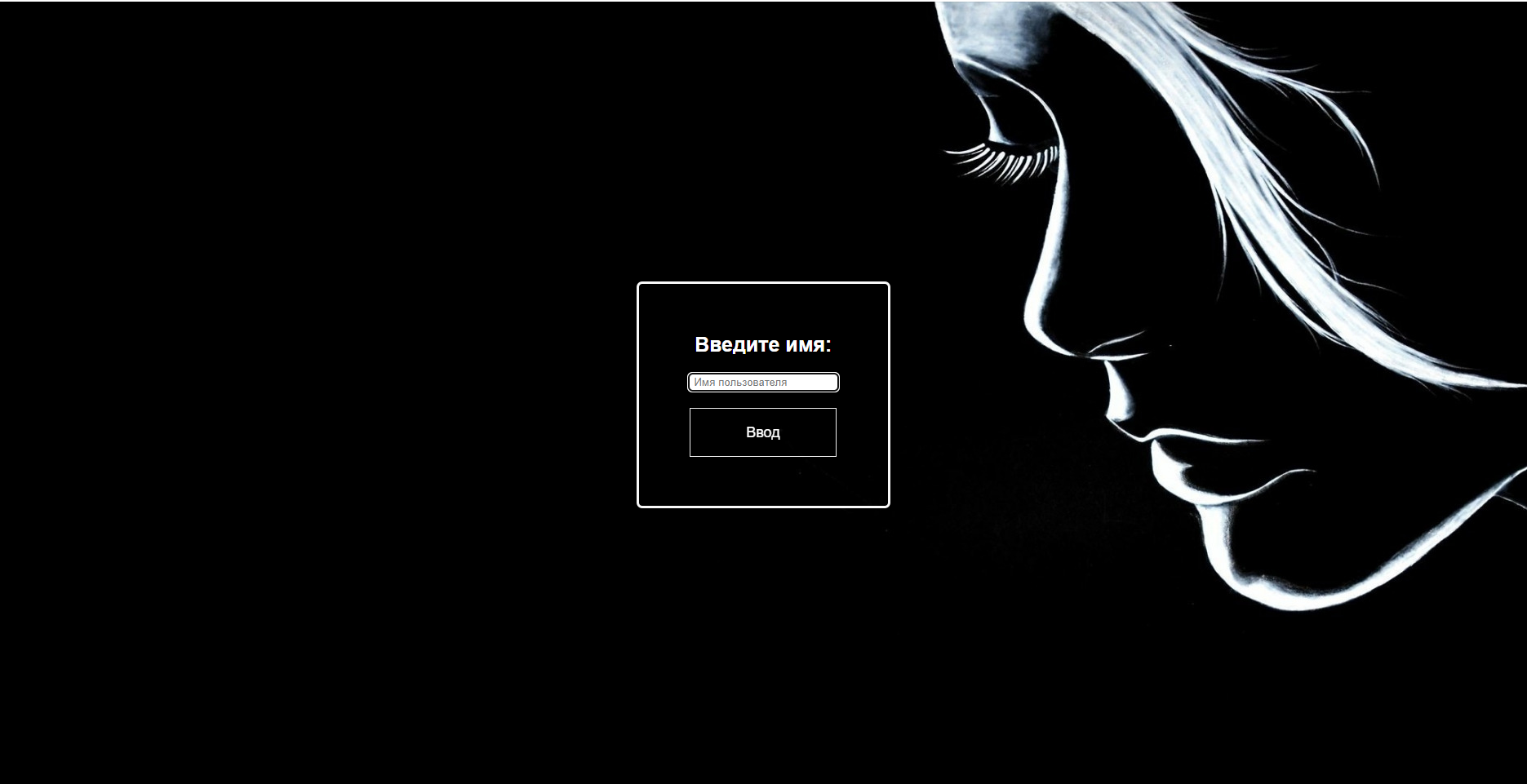
****

Рис 1 – Начальная страница.

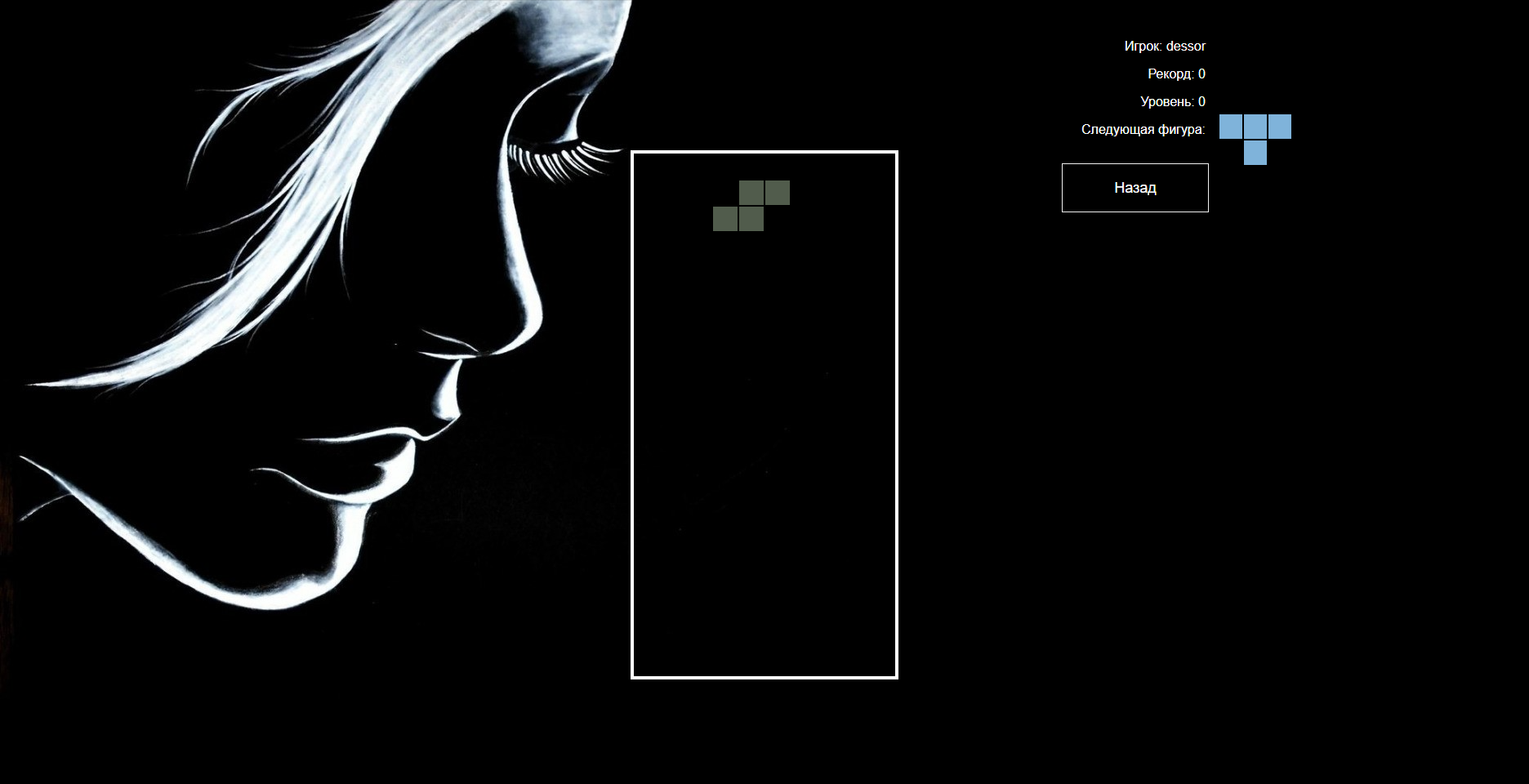


Рис 2 – Главная страница.



Рис 2 – Последняя страница.

## Выводы.

Была изучена работа *web*-сервера *nginx* со статическими файлами и создание клиентских *JavaScript* *web*-приложений. Получены практические навыки работы с *JavaScript.*